Neural Networks for Mass Regression for $HH \to bb\tau\tau$

A Thesis Presented

by

Storm Lin

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Master of Arts

in

Physics

Stony Brook University

May 2023

Abstract of the Thesis

Neural Networks for Mass Regression for $HH \to bb\tau\tau$

by

Storm Lin

Master of Arts

in

Physics

Stony Brook University

2023

The value of the Higgs boson self-coupling is predicted by the Standard Model, but it is still largely unconstrained experimentally and thus being actively studied at the ATLAS experiment at the LHC. The $HH \to bb\tau\tau$ decay channel is one of the most sensitive probes for studying this property. Regressing the di-Higgs invariant mass (m_{HH}) for events in this channel and understanding its distribution are important to improving the determination of the Higgs self-coupling. This thesis discusses recurrent, dense, and mixture density neural network models for predicting m_{HH} from other event variables. These new models demonstrate significant improvement in both accuracy of m_{HH} predictions and ability to separate m_{HH} distributions for different values of the self-coupling constant when compared to the currently adopted Missing Mass Calculator method for m_{HH} estimation in this channel.

Contents

Acknowledgements

I would first like to thank the two professors who were my primary advisors in this research, Giacinto Piacquadio and Quentin Buat. I would also like to thank Chris Pollard for his previous work and advice relating to neural networks at ATLAS. I would additionally like to acknowledge Jeff Backus, whose previous work on this topic was part of the basis for this research. I am also grateful to the members of the ATLAS $HH \to bb\tau\tau$ group and the members of the Stony Brook University ATLAS group for providing resources and feedback for this work. I would like to thank the National Science Foundation and the Research Foundation for the State University of New York for supporting this research financially.

Chapter 1

Introduction

Since its discovery in 2012, the Higgs boson has brought many answers and new questions for particle physics. The ATLAS experiment, one of the two experiments involved in the discovery, has since been involved in many other searches studying the properties of the Higgs. One particular feature of the Higgs boson in the Standard Model is its self-coupling, and multiple studies have been done to attempt to determine its self-coupling constant and whether or not it differs from the Standard Model expectation [1].

In order to probe the self-coupling, the production of Higgs boson pairs must be measured. One of the most sensitive decay channels for studying pairs of Higgs bosons is the $HH \to bb\tau\tau$ channel. An important challenge of the analysis in this channel is that this decay process produces neutrinos not visible to the ATLAS detector. This means that not all of the 4-momenta of the decay products of the Higgs bosons are directly observed by the detector. However, knowing the distribution of the total di-Higgs invariant mass m_{HH} is important to being able to determine the value of the self-coupling constant. As such, the currently-used Missing Mass Calculator (MMC) was developed to predict the missing mass of neutrinos based on other event variables to be able to account for the total m_{HH}.

However, the MMC has its weaknesses. It was originally built for analysis of $Z/H \to \tau\tau$, so it is not as accurate for $HH \to bb\tau\tau$ events. It also has a significant chance to fail for events with back-to-back τ leptons, resulting in estimated m_{HH} values that are significantly too low. As such, it is important to look for alternative methods that may be able to predict m_{HH} for this channel more accurately.

This research studies a new technique for regressing the invariant mass of the di-Higgs system in $HH \to bb\tau\tau$ events in the channel where both τ leptons decay hadronically using neural networks. By training on a large amount of Monte Carlo simulation event data, the neural network-based machine learning

system is able to develop a model that predicts the m_{HH} observable based on the other event variables. This paper discusses recurrent, dense, and mixture density neural networks, three different types of neural network regression models that can be used for this task. It details the methods by which these models are set up, trained, and evaluated. Additionally, it discusses promising new results that indicate more accurate predictions of m_{HH} in the $bb\tau\tau$ channel as well as an improved capability for distinguishing different values of the Higgs boson self-coupling constant with this technique when compared to the baseline MMC technique.

Chapter 2

Research Background Information

2.1 The LHC and the ATLAS Experiment

This research was done using simulations and data from the ATLAS particle physics experiment. Specifically, the data used in this research comes from Run 2 of ATLAS, which was gathered between 2015 and 2018 at a collision center-of-mass energy of $\sqrt{s} = 13$ TeV. ATLAS is a general-purpose particle detector at the Large Hadron Collider (LHC). The LHC is the world's largest and most powerful particle accelerator, located at the European Council for Nuclear Research (CERN) accelerator complex near Geneva, Switzerland. The LHC is a 27-km ring of superconducting electromagnets that is used to collide beams of protons [2].

The ATLAS detector [3] is centered around a collision point of the LHC and covers almost the entire solid angle around it. It is composed of a number of detector subsystems that are used to identify particles created by collision events and measure their momenta and energies. The inner detector is located within a 2 T axial magnetic field and is comprised of silicon pixel and strip detectors with straw-tube detectors outside and is used to measure the trajectories and momenta of charged particles like electrons. Outside of this, there is a liquid argon calorimeter system that has barrel and endcap components. The outermost layer of the detector is the muon spectrometer, which measures the trajectory of muons in the field of its magnets. ATLAS uses a two-level trigger system to determine which events are of interest and should be recorded. This system has a hardware component and a high-level software component which in combination reduce the LHC's 40 MHz proton bunch crossing rate to a final recorded event rate of about 1 kHz [4].

Events at the ATLAS detector are described in a particular coordinate system. The origin of the system is at the center of the detector, which is also approximately the center of the interaction region. The x-axis points towards the center of the LHC ring, the y-axis points upwards, and the z-axis points along the beam line. Points in space are described in terms of the distance from the z-axis r, the azimuthal angle around the z-axis ϕ, and the pseudorapidity η. The pseudorapidity is a function of the polar angle θ:

$$\eta = -\ln \tan \frac{\theta}{2} \tag{2.1}$$

2.2 The Standard Model and the Higgs Boson Self-Coupling

The Standard Model (SM) of particle physics is a well-established physical theory that describes the elementary particles that make up the universe and the fundamental forces by which these particles interact. The particles it describes are the six quarks in three generations, the six leptons in three generations, the photon which mediates the electromagnetic force, the gluon which mediates the strong nuclear force, the W and Z bosons which mediate the weak nuclear force, and the Higgs boson. While the SM does not describe all the physical phenomena known to science, such as gravitational force, it is the most accurate theory of elementary particle physics that exists today. Since its development in the 1970s, predictions based on the SM have been precisely supported by many experiments.

The SM particle of most interest in this research is the Higgs boson. The Higgs field, predicted in 1964, is the field that gives elementary particles their mass through their interactions with it. Confirmed in 2012 by the ATLAS [5] and CMS [6] collaborations, the Higgs boson is a particle excitation of this field. It is a spin-0 boson with a mass of about 125 GeV. Because the Higgs was confirmed relatively recently compared to the other SM particles, there are many unknowns about it, and a number of its properties are still being actively studied.

One property of the Higgs boson that is currently being studied is its self-coupling parameter λ_{HHH}. The Higgs boson self-coupling results in a trilinear and a quartic self-interaction Feynman diagram. The first and more likely of the two results in a Higgs boson decaying into two Higgs bosons. The decaying Higgs boson must be necessarily virtual with invariant mass $m_{HH} \geq 2m_H$. This process occurs with a coupling strength proportional to λ_{HHH}. Another way to describe the coupling is with the constant κ_λ, where $\kappa_\lambda = 1$ represents

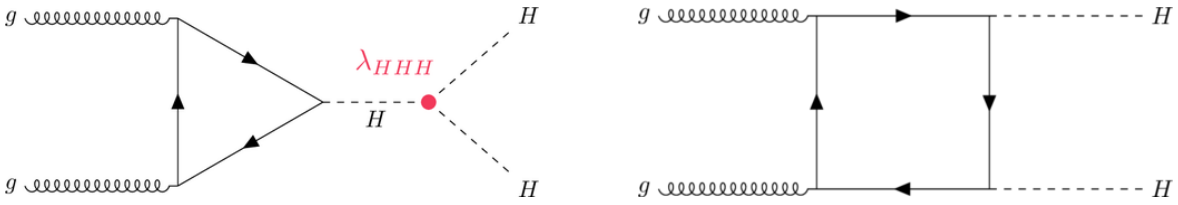


Figure 2.1: Feynman diagrams for the two leading-order processes for non-resonant production of a pair of Higgs bosons through gluon-gluon fusion. On the left, the 'triangle diagram' is a process that is sensitive to the Higgs self-coupling λ_{HHH}. On the right, the 'box diagram' is not sensitive to this self-coupling. Figures from [4].

the SM-predicted coupling strength, and other values of κ_λ represent couplings that are proportionally stronger or weaker than the SM prediction. It is not yet determined whether the observed Higgs self-coupling strength matches the SM prediction, but it has been experimentally bounded by ATLAS to be in the $-0.4 < \kappa_\lambda < 6.3$ interval at the 95% confidence level [1].

Studying the Higgs boson self-coupling begins with considering the possible processes which could produce a pair of Higgs bosons at ATLAS. This research focuses on the production of two Higgs bosons non-resonantly through gluon-gluon fusion (ggF). ggF is a process by which a particle, such as a Higgs boson, is produced from the interaction of two gluons in a proton-proton collision. The SM predicts that ggF accounts for about 90% of the total non-resonant production of pairs of Higgs bosons [4]. Non-resonant production means that there is no intermediate high-mass particle beyond the SM created that then decays into the two Higgs bosons. Figure 2.1 shows the leading-order Feynman diagrams for the production of two Higgs bosons this way. These Feynman diagrams interfere destructively, resulting in a very small SM cross-section for HH production, approximately one thousand times smaller than for single Higgs production. Given a cross-section of about 31 fb, the number of HH events expected over the full Run 2 dataset of $L = 140$ fb^{-1} is about 4,340 [7].

2.3 The $HH \to bb\tau\tau$ Channel

There are a number of different channels into which two Higgs bosons can decay after they are produced. This research focuses specifically on the $bb\tau\tau$ channel, where one of the Higgs bosons decays into a b quark and a b antiquark and the other decays into a τ^- lepton and a τ^+ lepton. The $bb\tau\tau$ final state comprises a substantial 7.3% of SM decays of HH, and the backgrounds in this channel are relatively low, making it one of the most sensitive channels for

studying HH [4]. This sensitivity is the reason it was chosen for this research. In addition, this research focuses on the hadronic channel for this decay, where both of the τ leptons decay into hadrons, which accounts for about 42% of all $\tau\tau$ pair decays and is the channel with the least amount of neutrinos [8].

Chapter 3

Objectives and Previous Methods

3.1 Regression of m_{HH} from Event Variables

An important task that was approached in this research was the regression of the invariant mass of the pair of Higgs bosons (m_{HH}) in an $HH \to bb\tau_{\text{had}}\tau_{\text{had}}$ event from other known event variables as precisely as possible. While the detector is able to measure the visible mass of the decay products, there is also a part of the total invariant mass that becomes missing mass in the form of neutrinos that the detector does not see. However, the most likely value of such missing mass, and thus of the total m_{HH}, can be predicted based on the values of the observables, such as visible τ and b-jet 4-momenta and missing transverse energy, that can be measured by the detector.

m_{HH} is a highly important variable for separating the signal from the background not only in non-resonant analysis but also in the resonant search analyses in the $X \to HH \to bb\tau\tau$ channel [4], so improving the capability to reconstruct this quantity could lead to better results for a number of different analyses. However, separating double-Higgs events from the background is only the first step. In order to constrain the value of the Higgs boson self-coupling, we need to separate the 'triangle diagram' from the 'box diagram' that is not sensitive to κ_λ (see Figure 2.1).

3.2 m_{HH} Distribution and the Self-Coupling Constant κ_λ

The reason that improving methods for regressing m_{HH} is important to understanding the Higgs self-coupling is that the distribution of m_{HH} depends on the coupling constant κ_λ. When many events are combined into a histogram, both the total number of $HH \to bb\tau\tau$ signal events as well the shape of the distribution of their m_{HH} are affected by κ_λ. For example, at $\kappa_\lambda = 10$ there will be many more signal events and the m_{HH} distribution will skew more towards low values of m_{HH} compared to $\kappa_\lambda = 1$. This can be seen in Figure 3.1. The reason for this is that the 'triangle diagram', which is proportional to κ_λ, results in a softer m_{HH} distribution compared to the 'box diagram'. Being able to better reconstruct the m_{HH} for individual events will also allow a more accurate reconstruction of the overall distribution of m_{HH}. With a better reconstruction of the m_{HH} distribution of experimentally observed events, tighter constraints can be imposed on the theoretically-predicted distributions that vary with κ_λ value, and ultimately tighter bounds will be able to be set on the true value of κ_λ.

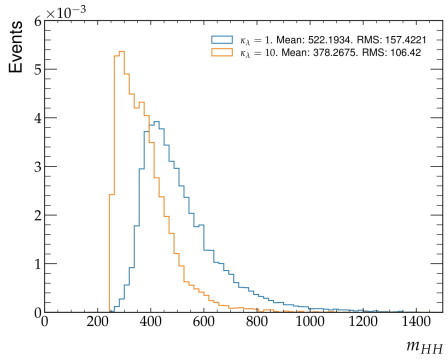


Figure 3.1: Overlaid true m_{HH} distributions for $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$. The legend displays the mean and root-mean-squared distance from the mean for both distributions. The $\kappa_\lambda = 10$ distribution has been scaled down to have the same integral as the $\kappa_\lambda = 1$ distribution for this plot, but it actually contains about 12 times as many events.

3.3 Missing Mass Calculator

The method currently used in ATLAS analyses for calculating the m_{HH} of $HH \to bb\tau\tau$ events is the Missing Mass Calculator (MMC) [9]. The MMC

calculates a prediction for the 4-momentum of the two τ leptons, including $m_{\tau\tau}$ and $\boldsymbol{p}_{\tau\tau}$, based on the 4-momentum of the visible part of the hadronic τ jets and the total missing transverse momentum. This information is then used along with the 4-momentum of the b quarks in order to obtain the total m_{HH}. In doing this, the MMC makes the assumption that all of the missing transverse momentum is from neutrinos produced when the τ leptons decay [4].

One weakness of the MMC is that there is a portion of events for which the MMC fails. This results in the predicted m_{HH} for the event being very low, and typically means that the event is discarded from the analysis. If another method is able to make reasonable m_{HH} predictions for these events where the MMC fails, then future analyses using this method will no longer have to cut out these events.

When developing methods to improve m_{HH} prediction, some key goals are thus obtaining more accurate predictions overall, avoiding making assumptions about where the missing mass goes, for example, taking into account that some of the missing energy also comes from neutrinos from b quark decays, and being functional for a more general class of events than the MMC.

Chapter 4

Methodology

4.1 Neural Networks

Neural networks are a type of machine learning artificial intelligence system made up of connected nodes called neurons [10]. Each node has a number of input connections and produces an output based on a linear combination of the input values with a set of coefficients called trainable parameters specific to that node. The linear combination is then passed through a function called an activation function that introduces nonlinearity in the output of the node. The type and number of neurons and their input and output connections to other neurons form the architecture of the network, which can be tuned to get better prediction performance. For example, networks that are too large or complex may take an inhibitively long time to train and may overfit the training data while networks that are too small or simple may fail to capture some aspects of complicated variable relationships.

Neural networks are trained using a large amount of paired input and output data in order to learn to predict the output variables from the input variables. A loss function is used to assess how well the model predicts the output based on the input, with lower values corresponding to more accurate predictions. During training, the parameters for each node are iteratively optimized by a gradient descent algorithm to minimize this loss function. This is done over the course of many training epochs, where the training repeats over the data until the model converges.

In addition to the trainable node parameters, there are also network hyperparameters, numbers that are set by the creator during model design. These parameters include the settings for the learning algorithm. The most important of these is the learning rate, which controls how fast the trainable parameters change each epoch. The model's convergence speed and the final

accuracy are very sensitive to the learning rate. Another training hyperparameter is the learning rate decay, controlling how the learning rate decreases from epoch to epoch in order to better assure convergence. Over the course of this research, for each network type and architecture used, the learning rate and other hyperparameters were carefully selected through testing models with a variety of different parameter values and selecting those which resulted in a low final loss and smooth loss convergence.

Neural networks can be used both to regress the value of numerical variables and to classify data points into discrete categories. In this research, regression neural networks were implemented in the Python programming language using the Keras deep learning API [11]. The networks were used to predict the invariant mass of the two Higgs bosons, m_{HH}, of events based on a set of sixteen event variables that were used as inputs to the network. These input variables are listed in 4.1. Previous work on this topic has used other sets of input variables, but comparison of model performance has shown this combination to be the most effective. Earlier models represented the transverse jet momenta as a total transverse momentum and an angle in the x-y plane, but testing showed better model performance when representing these momenta in Cartesian form with x and y components. It was also tested if converting the jet pseudorapidity to a z-momentum component would be beneficial, but this was found not to improve the performance. Additionally, the rotational invariance of the physics in the x-y plane allows all events to be rotated so that their missing transverse momentum lies along the x-axis, simplifying the input variable space by eliminating the need for having both x and y components for missing transverse momentum, eliminating one redundant degree of freedom and improving model performance. The output variable for the networks was m_{HH}/m_{vis}, the ratio of the total mass of the system to the visible mass. This output was chosen because previous work showed that models produced more accurate predictions when the network's output target was m_{HH}/m_{vis} rather than m_{HH} [12].

	Variables
Event-Level	Missing transverse energy Missing transverse energy significance
For the visible decays of each of the two τ jets	x-momentum component p_x y-momentum component p_y Pseudorapidity η
For each of the two b jets	x-momentum component p_x y-momentum component p_y Pseudorapidity η Mass m

Table 4.1: The sixteen event variables used as inputs for the neural network models. All events are rotated such that their missing transverse momentum lies along the x-axis. The pseudorapidity is defined as $\eta = -\ln(\tan(\theta/2))$ where θ is the azimuthal angle from the beam line.

4.2 Monte Carlo Simulation Event Generation and Selection

The event data used to train the neural networks is simulated data created by Monte Carlo (MC) event generators. These generators are used to simulate events like those that are observed at the ATLAS detector. They generate proton-proton collision events based on theoretical models, then simulate how the detector will respond to the events and how the events will be reconstructed from the detector readout data [13].

MC events are simulated for a number of different processes, and there are samples of events for each type of process. The four Monte Carlo event samples used for model training in this research come from the v. 15-01 bbtt_hh stream CxAOD files from the MC16a/d/e campaigns. These campaigns correspond to different periods of Run 2 LHC data-taking. There are two $HH \to bb\tau_{\text{had}}\tau_{\text{had}}$ signal samples: one simulated for $\kappa_\lambda = 1$ and one simulated for $\kappa_\lambda = 10$, and there are two background samples: $Z \to \tau\tau$ and tt. The $Z \to \tau\tau$ background sample was assigned a sample weight of 2 during the model-training process while the other three samples were each given a training weight of 1.

Not all of the events in each of the MC samples were used. Selection criteria were applied to the events in order to obtain the types of events that are of interest when studying the $HH \to bb\tau_{\text{had}}\tau_{\text{had}}$ channel while rejecting background-like events. Training sample events were selected according to a set of truth-guided criteria. That is, the selection was done based on the par-

Sample Description	Dataset ID	Number of Raw MC Events after Selection	Number of Expected Data Events
$HH \rightarrow bb\tau_{\text{had}}\tau_{\text{had}}$ ($\kappa_\lambda = 1$) Signal	600023	96,388	25
$HH \rightarrow bb\tau_{\text{had}}\tau_{\text{had}}$ ($\kappa_\lambda = 10$) Signal	600024	56,464	288
$Z \rightarrow \tau\tau + \text{jets}$ Background	364128-364141	59,877	3,036
tt Background	410470-410471, 410644-410647, 410658-410659	30,415	4,473

Table 4.2: The Monte Carlo samples used for model training, the raw number of events they contain, and the number of events expected in data with $L = 140$ fb^{-1}.

ticles that would be reconstructed from the detector readout, and then these reconstructed particles are matched to the particles that were truly present in the MC event. The criteria were that the events have exactly two hadronically-decaying τ leptons and exactly two b quark jets. Additionally, for the signal, events were selected on the criterion that they have exactly two truth Higgs bosons. Identifying information for the MC samples used as well as the number of events used from each sample can be found in Table 4.2.

The variable of interest, m_{HH}, only truly represents the mass of two Higgs bosons for the signal events, but a definition of the variable must be set forth for all events. For signal MC events, the m_{HH} is defined as the sum of the 4-momenta of the two truth τ leptons and the two truth b quarks. For the $Z \rightarrow \tau\tau$ background events, the m_{HH} is defined as the sum of the 4-momenta of the two truth τ leptons and the two reconstructed b jets as there is no truth b quark 4-momenta information for these events. For the tt background events, the m_{HH} is defined as the sum of the 4-momenta of the two truth τ leptons and the two truth b quarks with some exceptions when this is not possible. If there are three truth τ leptons, only the two τ leptons with the highest `MC status` variable value are used. For events with only one truth b quark, the reconstructed b jets were used instead. tt events with only one truth τ lepton were not considered.

4.3 Training-Testing Split

When using machine learning techniques, it is important to set aside a part of the available data to be used for evaluating how the neural network will perform on unseen data. This is called a training-testing split. The network is trained using only the training part of the data. After the network has converged, its performance is evaluated using only the testing data, which has not been seen by the network up to this point.

For the neural networks in this research, the split used was that 2/3 of the events were used for training and 1/3 were reserved for testing. This split was done based on the value of each event's MC event identification number modulus 3. For some of the models in this research, this split was only done once, resulting in a model which is a single neural network trained on 2/3 of the MC events and evaluated on 1/3 of the events. These models will be referred to as non-folded models.

The other models, referred to as folded models, consist of three different neural networks. These models were trained in three folds in order to make full use of all of the available MC events for training. The way this folding works is that one network was trained only on events with event number modulus 3 not equal to 0 and tested on events with event number modulus 3 equal to 0, and then a second network was prepared with the same procedure but for modulus equal to 1 and a third for modulus equal to 2. When evaluating a folded model overall on all of the MC events, each MC event was predicted only by the network that did not see that event as part of its training in order to get the most accurate picture of how the model will perform when used with real detector data.

4.4 Recurrent Neural Network Models

One type of neural network model used in previous work on this topic was the recurrent neural network (RNN) [14]. This type of network allows nodes to be connected in a cyclic temporal sequence where the output of a node can later affect that same node's inputs. This is done by allowing the network to store its own data that it writes to and reads from every time it is run, allowing past runs of the model to influence future runs. In practice, this allows the network to process variable-length sequences of inputs. Previous work has demonstrated that RNN models are able to outperform the MMC at the task of predicting m_{HH} [12].

The goal of this current research was to explore other neural network architectures to see if they are able to do even better at this task. One of the

motivations to do this is that RNN models take a much longer time to train compared to many other architectures for a comparable number of nodes and trainable parameters. Another reason to seek alternative architectures for this problem is that the key feature of an RNN is its recursive nature that allows data points to influence the network's calculations for subsequent data points, but this feature is not specifically motivated for this regression problem as it is not expected that particle collision events are correlated with each other in the sequence in which they are processed.

4.5 Dense Neural Network Models

The first neural network architecture primarily explored in this research was a deep dense neural network (DNN). The network is deep, meaning that there are multiple hidden layers of neurons in between the input and output layers. Dense networks, also known as fully-connected networks, are neural networks where every neuron in each layer is connected to every neuron in the previous layer and every neuron in the next layer. This type of architecture is simpler than a recurrent network. The advantage of a simpler architecture is that the training times are significantly shorter even for a model with many more nodes, making it easier to iterate on the model and fine-tune its architecture and hyperparameters. The loss function used when training these models was the squared error, that is, the difference between the predicted and true m_{HH}/m_{vis} squared.

4.6 Mixture Density Network Models

The other neural network technique that was a focus of this research was the mixture density network (MDN). An MDN differs from a DNN only in its output. Rather than outputting a single value for the regressed m_{HH}/m_{vis} of an event, it outputs a set of variables that parameterize a predicted distribution of m_{HH}/m_{vis} for events with given input variables. In the specific case used in this research, this predicted distribution was assumed to be Gaussian, and it was parameterized in terms of 2 variables: the mean and the standard deviation of the Gaussian.

In order to optimize this 2-output network, a different loss function was needed. This research used a typical loss function for MDNs, the negative log-likelihood (NLL). This loss function's value is equal to the negative of the logarithm of the probability distribution predicted by the network, evaluated at the true m_{HH}/m_{vis} value. For a Gaussian distribution, like the one assumed

in this research, the probability distribution is:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \tag{4.1}$$

where x is the true value of m_{HH}/m_{vis}, μ is the mean of the predicted distribution, and σ is the standard deviation of the predicted distribution.

Other than the differences in its output, the nodes in the MDN architecture are simply connected as in the DNN architecture, and the layout is described in Figure 4.1. The activation function for the nodes in the 6 intermediate hidden layers is the Rectified Linear Unit (ReLU), which is linear for a positive input and 0 for a negative input. The activation function for the output layer is linear. As this network is also dense, each node is connected to each other node in the previous layer and in the next layer. This type of MDN maintains the benefits gained from the switch from RNNs to DNNs in that the simple dense node architecture allows for relatively fast training times even with a large number of nodes and trainable parameters. The number of nodes for the MDN network was selected so that the total number of trainable parameters for the network, 48,266, is on the same order of magnitude as, but slightly less than, the number of events the network will see during training, which is either about 101,948 or about 161,919 depending on whether the network is being trained only on the two signal samples or on all four samples. This allows the network to be large enough to model complex relationships between variables while also minimizing the potential issue of overfitting the training data.

A notable advantage of the MDN architecture compared to the single-output DNN architecture is the added information of the second output. The MDN's first output, μ, is the mean of the predicted distribution for m_{HH}/m_{vis} and can be interpreted as the best estimate of the model for the true m_{HH}/m_{vis} value. The second output, σ, is the standard deviation or width of the predicted distribution and can be interpreted as an uncertainty in the best estimate for m_{HH}/m_{vis}. This research has demonstrated that the MDN's μ output is as good of a prediction of m_{HH}/m_{vis} as the single output value of the DNN, meaning that the σ output of the MDN is purely additional information gained from using this architecture. σ is then a potentially valuable piece of information not present in a single-output DNN that can be used to separate events based on whether they have high or low uncertainty in the predicted m_{HH}/m_{vis}. The ability to categorize events in this way can be used to improve the separation of signals with different underlying true m_{HH} distributions.

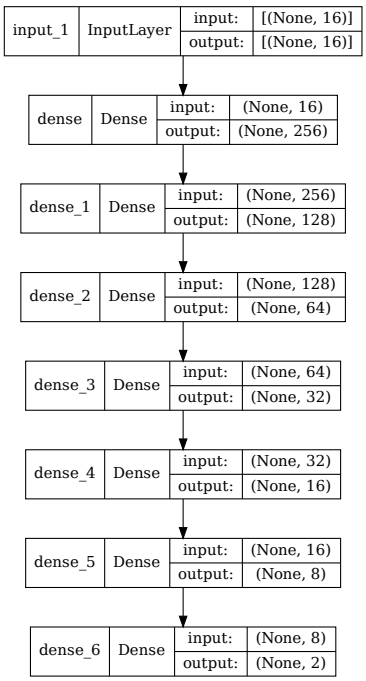


Figure 4.1: A visual representation of the node layout architecture for the MDN.

Chapter 5

Assessing Performance and Results

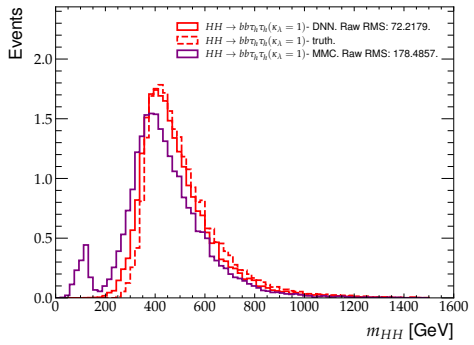
5.1 m_{HH} Distribution

The first way of assessing the performance of the neural network models is to look at how well the regressed m_{HH} distribution matches the true distribution. Three neural network models are presented here: a non-folded DNN model trained on all four MC samples, a folded MDN model trained on all four MC samples, and a folded MDN model trained only on the two signal MC samples. The first comparison that is made is overlaying the new network's distributions of predicted m_{HH} for each of the MC event samples with the MMC prediction and the truth. Then, the difference between the predicted and true m_{HH} for each MC event can be calculated and used to obtain a root-mean-square (RMS) error for both the network's and the MMC's predictions. The DNN, all-samples MDN, and signal-only MDN distribution comparisons are shown in Figures 5.1, 5.2, and 5.3, respectively. Additionally, the root-mean-squared errors in m_{HH} for each MC sample for each of these models are summarized in Table 5.1.

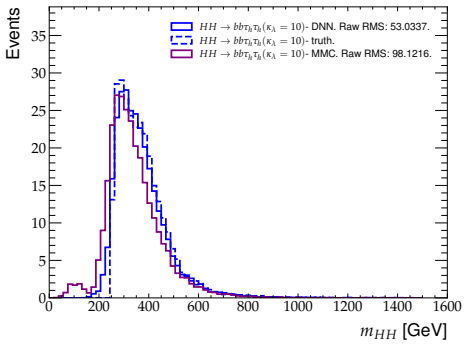
A noticeable feature of these distribution plots is that the MMC-predicted distribution consistently has a cluster of events at low masses below about 200 GeV. These correspond to events where the MMC calculation fails, which account for about 8.5% of the total number of events in these MC samples. It can then be seen that there is no such cluster of failed events for the neural network predictions, demonstrating one of the advantages of using neural networks for this regression. Another notable observation about these distributions is that the MDN trained on all four samples fits the true distribution for the background samples much better than the signal-only MDN, which

Model	Sample	RMS Error
MMC	$\kappa_\lambda = 1$ Signal	178.1372
	$\kappa_\lambda = 10$ Signal	97.5132
	$Z \to \tau\tau$ Background	123.512
	tt Background	164.5834
DNN	$\kappa_\lambda = 1$ Signal	72.2179
	$\kappa_\lambda = 10$ Signal	53.0337
	$Z \to \tau\tau$ Background	54.2985
	tt Background	86.405
All-Samples MDN	$\kappa_\lambda = 1$ Signal	75.68
	$\kappa_\lambda = 10$ Signal	56.1872
	$Z \to \tau\tau$ Background	52.7059
	tt Background	86.9514
Signal-Only MDN	$\kappa_\lambda = 1$ Signal	56.5971
	$\kappa_\lambda = 10$ Signal	43.1203
	$Z \to \tau\tau$ Background	146.6894
	tt Background	170.6501

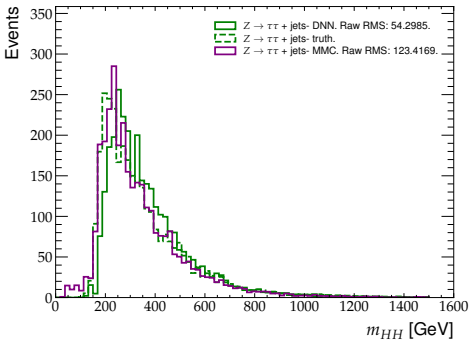
Table 5.1: The root-mean-squared error in m_{HH} for each MC sample for the different m_{HH} regression models.



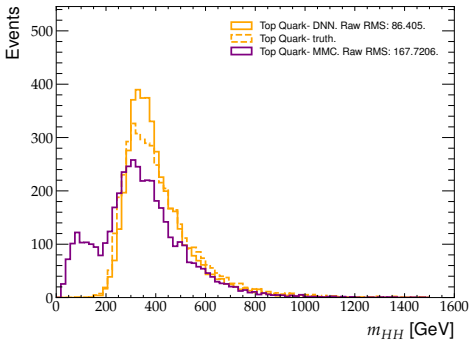
(a) $\kappa_\lambda = 1$ Signal



(b) $\kappa_\lambda = 10$ Signal



(c) $Z \to \tau\tau$ Background

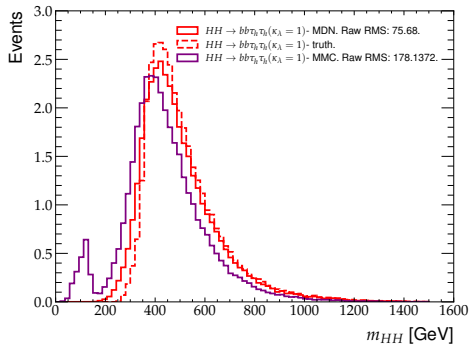


(d) tt Background

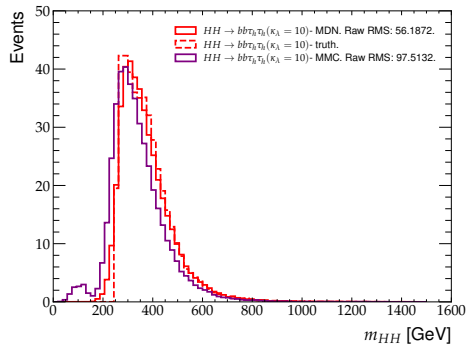
Figure 5.1: Regressed m_{HH} distributions for the DNN compared to the MMC and truth distributions for each of the four samples. Raw RMS here is the root-mean-squared error in m_{HH}.

makes sense as the all-samples MDN saw background events during training while the signal-only MDN did not. Except in the case of the signal-only MDN predicting for the background samples, the neural network predicted distribution's RMS error was always less than the RMS error for the MMC prediction. This is strong evidence that the neural network models are able to provide more accurate prediction for m_{HH} than the MMC.

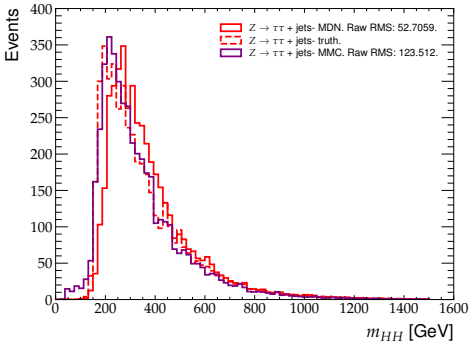
Another useful way to examine the accuracy of the predictions is to create plots of the distributions of predicted m_{HH} of each event divided by the true m_{HH} of that event. This distribution is expected to be roughly Gaussian with a mean equal to 1 if the predictions for m_{HH} are unbiased and a width corresponding to the fractional resolution, so a Gaussian will be fitted to the distributions for both the neural network and the MMC. The DNN, all-samples MDN, and signal-only MDN distribution of predicted divided by true m_{HH}



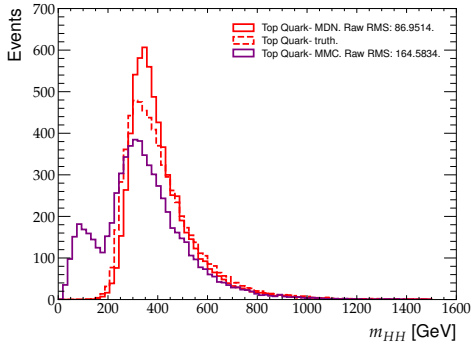
(a) $\kappa_\lambda = 1$ Signal



(b) $\kappa_\lambda = 10$ Signal



(c) $Z \to \tau\tau$ Background

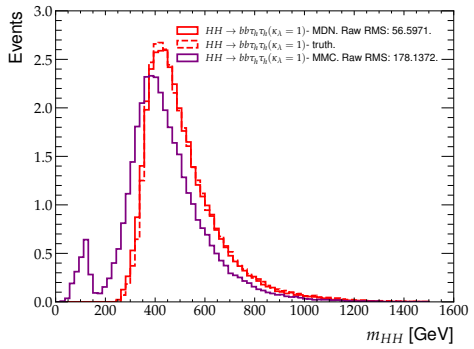


(d) tt Background

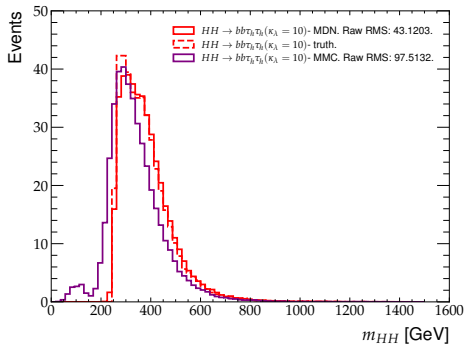
Figure 5.2: Regressed m_{HH} distributions for the MDN trained on all four samples compared to the MMC and truth distributions for each of the four samples.

and the Gaussian fits are shown in Figures 5.4, 5.5, and 5.6, respectively. The Gaussian fit information for each of these models is additionally summarized in Table 5.2.

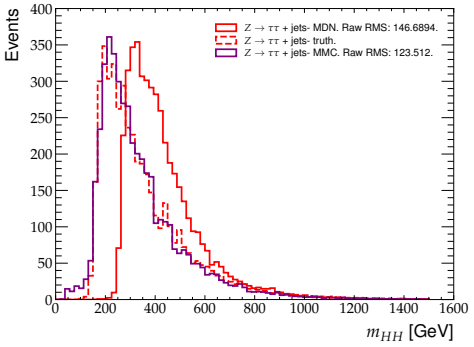
For these prediction-to-truth ratio plots, good model performance would be quantified as having fitted Gaussians with μ as close as possible to 1, indicating that the model is an unbiased predictor of m_{HH}, and σ as small as possible, indicating that most of the predictions are very close to the truth. Looking first at the two signal samples, the DNN and both MDN models clearly outperform the MMC in this regard. The neural network models all have μ within 0.03 of 1 while the MMC has a μ of about 0.93. Thus, the neural network models have substantially less biased m_{HH} predictions compared to the MMC. The neural network models all have $\sigma < 0.11$ while the MMC has a σ between 0.12 and 0.13, indicating that the neural network models are on average slightly



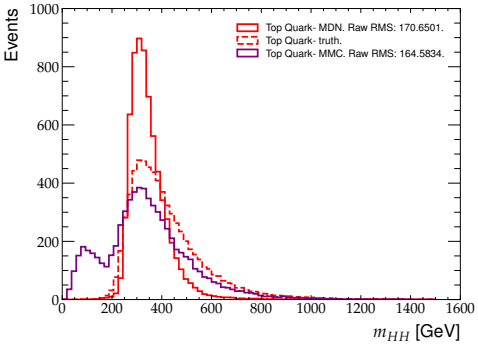
(a) $\kappa_\lambda = 1$ Signal



(b) $\kappa_\lambda = 10$ Signal



(c) $Z \to \tau\tau$ Background

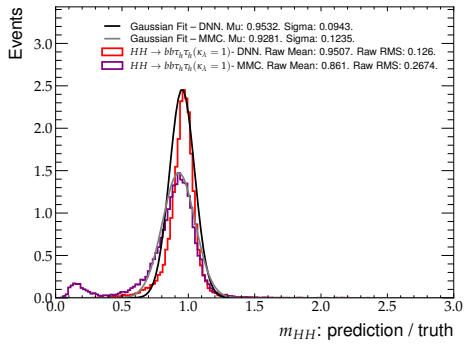


(d) tt Background

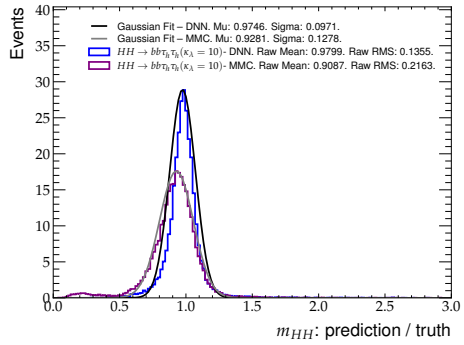
Figure 5.3: Regressed m_{HH} distributions for the MDN trained only on the two signal samples compared to the MMC and truth distributions for each of the four samples.

more accurate in their predictions. The MDN trained only on signal samples has a σ around 0.08, evidencing that a network trained on only signal samples will be able to give more accurate m_{HH} predictions for signal events.

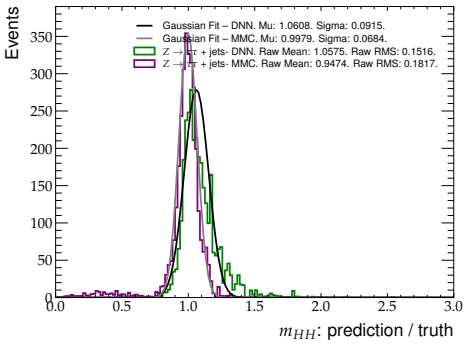
Looking at the two background samples, the first observation is that the MMC outperforms the DNN and MDN models for the $Z \to \tau\tau$ samples with both μ closer to 1 and smaller σ. This is an expected outcome because the MMC determines the most likely solution to the equation system based on probability density functions that come from $Z \to \tau\tau$ simulations, and it is thus very good at predicting m_{HH} for $Z \to \tau\tau$ events. For the top quark background, the DNN and the all-samples MDN outperform the MMC with both μ closer to 1 and smaller σ. However, the signal-only MDN does not. This makes sense, as a model trained without ever seeing the background samples has a much more difficult time making accurate predictions for background



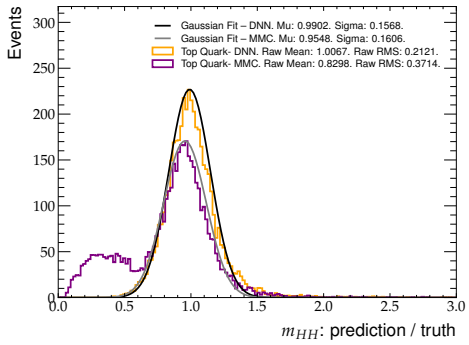
(a) $\kappa_\lambda = 1$ Signal



(b) $\kappa_\lambda = 10$ Signal



(c) $Z \to \tau\tau$ Background



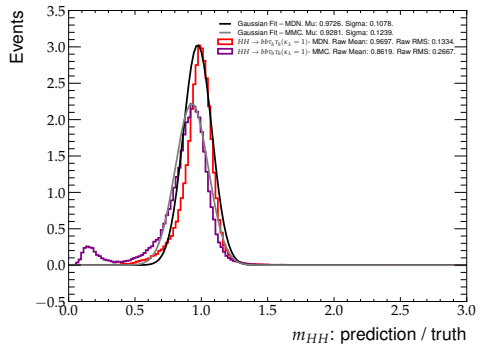
(d) tt Background

Figure 5.4: Distributions of predicted divided by true m_{HH} for the DNN compared to the MMC for each of the four samples. A Gaussian curve is fitted to each of the DNN and MMC distributions. This Gaussian is parameterized by its mean μ and standard deviation σ.

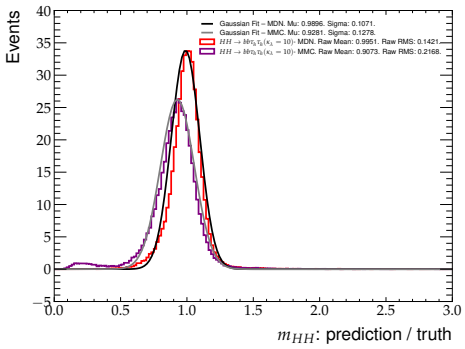
events. Another important observation is that the MDN trained on all samples has very similar performance to the DNN trained on all samples. This shows that MDN models are generally predicting m_{HH} at least as well as DNN models while also being able to add in their predicted event σ information. As such, from this point onward, all plots shown will focus on the MDN models.

Model	Sample	Gaussian Fit μ	Gaussian Fit σ
MMC	$\kappa_\lambda = 1$ Signal	0.9281	0.1239
	$\kappa_\lambda = 10$ Signal	0.9281	0.1278
	$Z \to \tau\tau$ Background	0.9964	0.0702
	tt Background	0.9536	0.1614
DNN	$\kappa_\lambda = 1$ Signal	0.9532	0.0943
	$\kappa_\lambda = 10$ Signal	0.9746	0.0971
	$Z \to \tau\tau$ Background	1.0608	0.0915
	tt Background	0.9902	0.1568
All-Samples MDN	$\kappa_\lambda = 1$ Signal	0.9726	0.1078
	$\kappa_\lambda = 10$ Signal	0.9896	0.1071
	$Z \to \tau\tau$ Background	1.0852	0.0975
	tt Background	1.0062	0.1575
Signal-Only MDN	$\kappa_\lambda = 1$ Signal	0.9901	0.0849
	$\kappa_\lambda = 10$ Signal	1.0114	0.0807
	$Z \to \tau\tau$ Background	1.0906	0.129
	tt Background	0.9077	0.1989

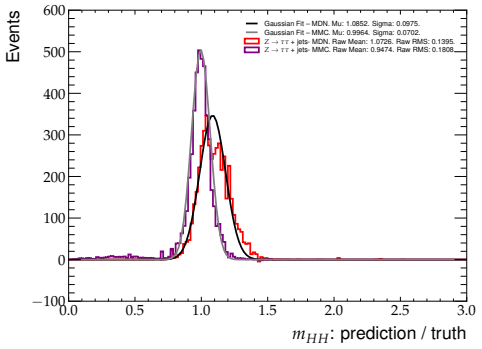
Table 5.2: The results of Gaussian fitting for predicted m_{HH} divided by true m_{HH} for the different m_{HH} regression models.



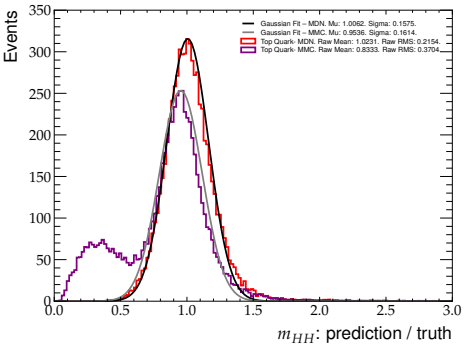
(a) $\kappa_\lambda = 1$ Signal



(b) $\kappa_\lambda = 10$ Signal

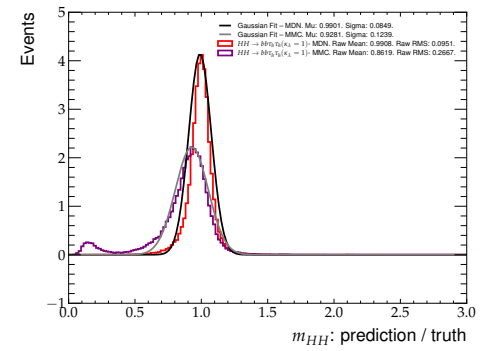


(c) $Z \to \tau\tau$ Background

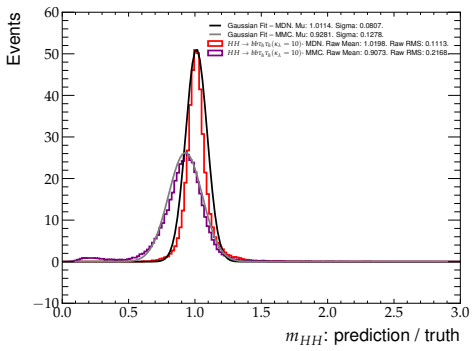


(d) tt Background

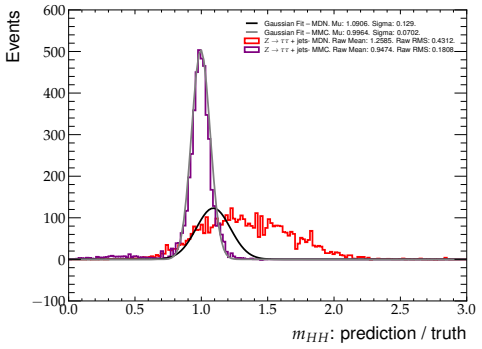
Figure 5.5: Distributions of predicted divided by true m_{HH} for the MDN trained on all four samples compared to the MMC for each of the four samples.



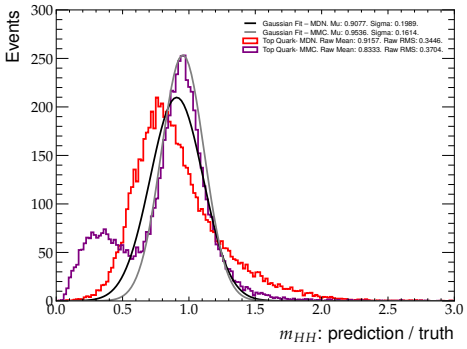
(a) $\kappa_\lambda = 1$ Signal



(b) $\kappa_\lambda = 10$ Signal



(c) $Z \to \tau\tau$ Background



(d) tt Background

Figure 5.6: Distributions of predicted divided by true m_{HH} for the MDN trained only on the two signal samples compared to the MMC for each of the four samples.

5.2 Separation Significance

One of the main goals of this research is to improve the ability to determine κ_λ based on the regressed m_{HH} distribution of the data. As such, it is an important part of evaluating the model to quantify how well the regressed m_{HH} distributions allow the separation of signals with different κ_λ values. In order to do this, one can look at the separation significance Z between two hypotheses '1' and '2', defined as:

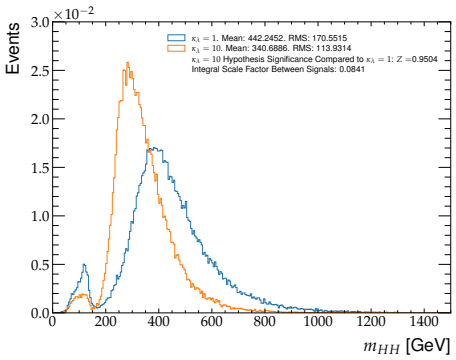
$$Z = \sqrt{2 \sum_i \log\left(\frac{f_2^i}{f_1^i}\right) f_2^i}, \tag{5.1}$$

where f_1 is the binned probability distribution function (PDF) of the first m_{HH} distribution and f_2 is the binned PDF of the second m_{HH} distribution. The sum is over the bins of the function, which have a width of 5 GeV in this research, excluding bins where either distribution is 0. This significance is valid for normalization to 1, but the separation significance for N events can be obtained by multiplying by $\sqrt{N}$. If the distributions were not normalized to unity and this separation formula was applied, the separation would be dominated by the very large difference in the total signal cross-section (approximately a factor of 12) between the $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$ distributions. However, in this work, the interest is in the additional separation from using differential distributions. Figure 5.7 shows the $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$ predicted m_{HH} distributions side-by-side for the MMC, the all-samples MDN, and the signal-only MDN, and the separation significance for each. These separation significances are also summarized in Table 5.3.

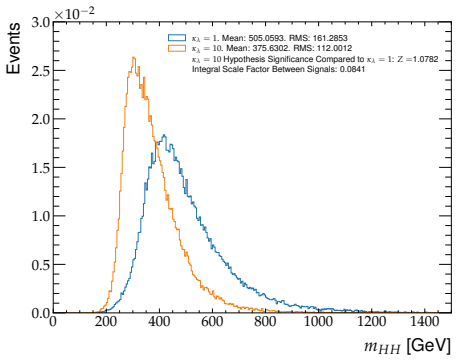
The separation significance values calculated in these plots show that there is a significant improvement of about 13% in the separation significance going from the MMC to the all-samples MDN and another significant improvement of about 17% when going from the all-samples MDN to the signal-only MDN. The MDN models are both able to separate the $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$ m_{HH}

Model	Separation Significance
MMC	0.9504
All-Samples MDN	1.0782
Signal-Only MDN	1.2661

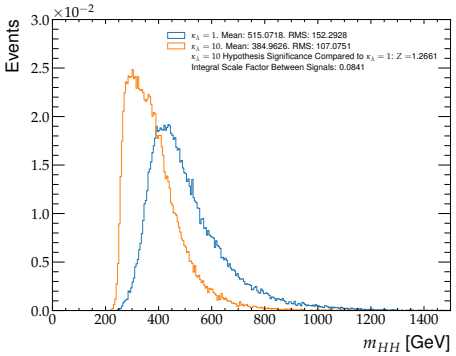
Table 5.3: The separation significances between the m_{HH} distributions for $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$ for different m_{HH} regression models.



(a) MMC



(b) All-Samples MDN

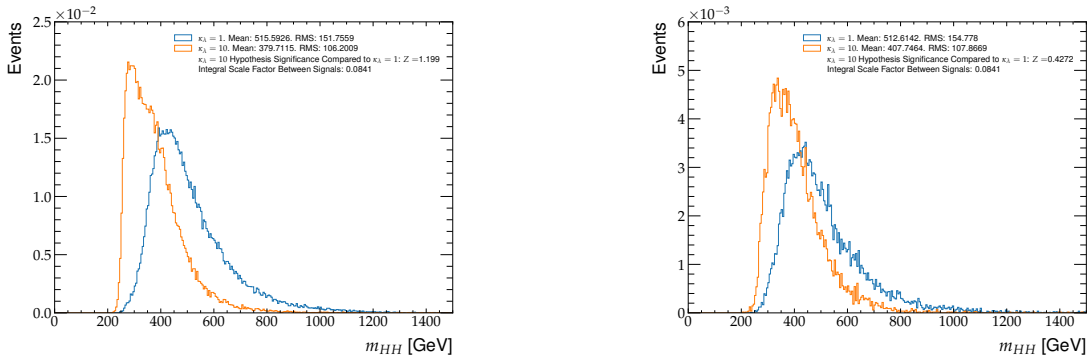


(c) Signal-Only MDN

Figure 5.7: Overlaid predicted m_{HH} PDFs for $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$. The legend displays the mean and root-mean-squared distance from the mean for both distributions, the separation significance, and the ratio of the total integral cross-sections of the two distributions if they were not normalized to PDFs. The first plot is for the MMC, the second plot is for the MDN trained on all four samples, and the third plot is for the MDN trained only on the two signal samples.

from each other substantially better than the MMC can, and the MDN trained only on the signal has a large additional improvement to separation compared to the MDN trained on all the samples. The primary interest of this research is to study κ_λ, so the ability to separate m_{HH} distributions of different κ_λ is highly important. Because of this, the MDN trained only on signal samples will be the focus of the plots going forward, as it is the model that achieves the best separation significance between the $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$ distributions.

MDN models provide a predicted distribution standard deviation σ for each event in addition to the predicted distribution mean μ. μ acts as a best



(a) $\sigma/\mu < 0.1$ (b) $\sigma/\mu > 0.1$

Figure 5.8: Overlaid predicted m_{HH} PDFs for $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$ for the MDN trained only on the two signal samples when events are separated based on their σ value. The left plot contains the events with $\sigma/\mu < 0.1$, and the right plot contains the events with $\sigma/\mu > 0.1$. The total separation significance in this case is $Z = 1.273$.

estimate of the true m_{HH}/m_{vis} value, while σ can be interpreted as a predicted uncertainty in this estimate. One way to utilize the event σ information added by MDN models is to categorize events based on whether they have high or low σ. In the case of this research, events were categorized based on if the MDN's predicted σ/μ for that event was less than or greater than 0.1, a number chosen because it is approximately the mean of σ/μ over all events in the samples. After events are placed into these categories, the separation significance calculation can be done again with the two new $\kappa_\lambda = 1$ distributions and two new $\kappa_\lambda = 10$ distributions. Note that when doing this, the two categorized distributions will sum up to a complete PDF with integral 1, but will not each be PDFs with integral 1 on their own. Once the separation significance for each of the categories is calculated, they can be summed in quadrature to get the total separation significance:

$$Z = \sqrt{Z_1^2 + Z_2^2}. \tag{5.2}$$

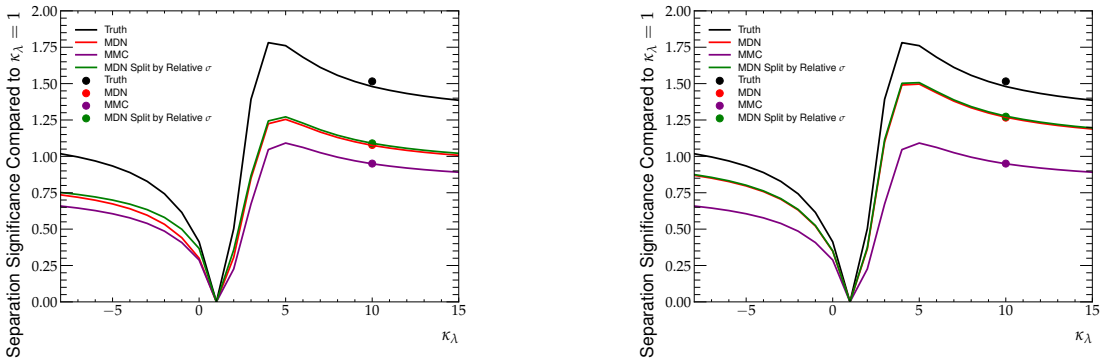
Figure 5.8 shows this being done for the MDN trained only on the signal samples, and demonstrates that the total categorized separation significance is slightly increased compared to the uncategorized separation significance. For the separation significance between the $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$ MC samples, event categorization based on σ improves the signal-only MDN's separation significance by about 0.54% from 1.2661 to 1.273.

5.3 κ_λ Reweighting

While MC samples only exist for $\kappa_\lambda = 1$ and $\kappa_\lambda = 10$ signals, it is important to also understand how well the m_{HH} distribution can be used to separate the $\kappa_\lambda = 1$ signal from signals for other values of κ_λ. To do this, one can use existing data that describes how to reweight the true $\kappa_\lambda = 1$ m_{HH} distribution to obtain the true m_{HH} distributions for other values of κ_λ. This data provides reweighting factors for each m_{HH} histogram bin that can be multiplied by the total weight of events in that bin to obtain the total weight of events in that bin for the new κ_λ value. This reweighting can also be applied at an event level. To do this, for each event in the $\kappa_\lambda = 1$ signal sample, look at which histogram bin the truth m_{HH} of that event would fall into, then apply the appropriate reweighting factor from the matching bin. This results in every event in the sample now having a new event weight that results in the truth m_{HH} distribution for the sample now matching the truth m_{HH} distribution for the κ_λ value of interest.

Using this reweighting technique, the separation between the $\kappa_\lambda = 1$ signal m_{HH} distribution and the m_{HH} distribution for any other κ_λ value can be computed. A scan can be constructed of the separation from different values of κ_λ, and this can be presented as a plot with κ_λ on the x-axis and separation on the y-axis. Such a plot can be used to compare the effectiveness of different m_{HH} reconstruction methods by looking at their ability to separate signals across a range of κ_λ values. Figure 5.9 shows this κ_λ separation scan plot for both of the MDN models.

These plots show that across a broad range of κ_λ values, the MDN models significantly outperform the MMC, achieving separation significance values substantially closer to the truth. Additionally, the signal-only MDN has significantly better separation significance than the all-samples MDN over the entire κ_λ range. This provides further evidence supporting the decision to focus on the signal-only MDN for its stronger distribution-separating ability. For both of the MDN models, splitting events into two categories based on their predicted σ increases the separation significance by an additional small amount. This shows that the new σ information gained by using an MDN rather than a DNN is useful for separating m_{HH} distributions for different κ_λ.



(a) All-Samples MDN

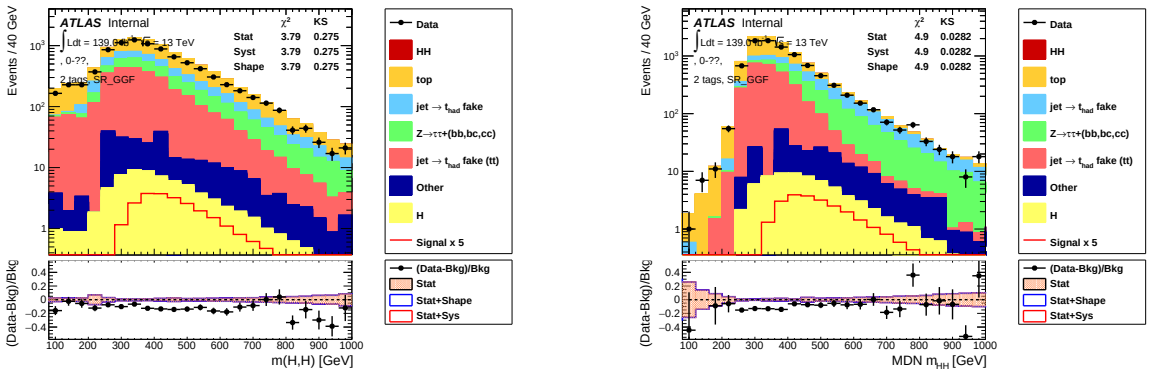
(b) Signal-Only MDN

Figure 5.9: Scans of separation significance compared to $\kappa_\lambda = 1$ as a function of κ_λ comparing the truth, MMC, MDN, and MDN split into two categories based on σ and then recombined. The data points at $\kappa_\lambda = 10$ represent the separation significances calculated with the $\kappa_\lambda = 10$ MC sample, not using a reweighted $\kappa_\lambda = 1$ sample. The left plot shows this scan for the MDN trained on all four samples while the right plot shows this scan for the MDN trained only on the two signal samples.

5.4 Incorporating this Model into a Run 2 Analysis

In order to better understand how these neural network models will perform on real data, the neural network-based m_{HH} prediction was implemented in the official analysis framework and added as a variable in CxAODReader. In fact, beyond the expected performance in the signal as predicted by the MC simulations, it is important to check that the new regressed m_{HH} observable is well-modeled in actual data. CxAODReader is a C++ analysis software that is used to run the last step of the event reconstruction, build analysis-level observables, and apply the analysis selection. A series of criteria are applied to reconstructed objects and event kinematics to construct histogram templates of each simulation sample as well as the data. Figure 5.10 shows the comparison between the combined MC samples and the ATLAS Run 2 data for the predicted m_{HH} distributions from the MMC and signal-only MDN in the signal region with two b-tagged jets and two hadronic τ leptons.

It can be seen in these plots that the MDN-predicted m_{HH} demonstrates good agreement between MC and data that is comparable to the agreement in the MMC-predicted m_{HH}. Similar to what was seen in the purely simulation-based m_{HH} distribution plots, there is an excess of events at low m_{HH} of events



(a) MMC (b) Signal-Only MDN

Figure 5.10: Histogram of predicted m_{HH} comparing the MC to the ATLAS Run 2 data. This histogram is of events with two tagged b-jets and τ leptons of opposite sign that were categorized as gluon-gluon fusion. The left plot shows this histogram for the MMC while the right plot shows this histogram for the MDN trained only on the two signal samples.

for which the MMC fails that is not seen for the MDN. These plots also show that the data matches the MC for the MDN about as well as it does for the MMC in the m_{HH} range where most events are found such that there is no showstopper in adopting the new MDN-based regression technique presented in this work in the next $HH \to bb\tau\tau$ analysis.

Chapter 6

Conclusion

This work has discussed the use of recurrent, dense, and mixture density neural networks for the task of regressing the m_{HH} of $HH \to bb\tau\tau$ events from other event variables. As shown by the results presented, these new neural network models are capable of outperforming the MMC at this task in several different ways. It was shown that for all of the MC simulation samples except for $Z \to \tau\tau$, neural network models were able to produce m_{HH} predictions that were more accurate than the MMC. The neural networks also do not have the cluster of failed-calculation events at low m_{HH} that the MMC has, meaning that potentially more events could be included in the analysis that were previously cut out due to MMC failure. When computing separation significance for the m_{HH} distributions for different values of κ_λ, the MDN models were shown to have achieved a higher separation significance compared to $\kappa_\lambda = 1$ than the MMC over the entire range of tested κ_λ values.

Additionally, it was shown that the σ information that is added when using an MDN model does not hurt the model's m_{HH} prediction accuracy. This σ information was shown to be useful when used to separate events into categories based on whether they have high or low σ. Doing this splitting and then combining the separation significance for the two categories results in a slight improvement in total ability to separate the m_{HH} distributions for different values of κ_λ compared to simply calculating the separation significance with all of the events without performing any categorization.

Another finding of this research is that MDN models trained only on $HH \to bb\tau\tau$ signal MC samples perform better at predicting the m_{HH} of signal events as well as at separating the m_{HH} distributions for different κ_λ values compared to MDN models trained on both signal and background event samples. However, this comes at the cost of being drastically less accurate at predicting the m_{HH} for background events.

There is still more to be done with the neural network techniques developed

in this research. While several different models and network architectures were tested throughout this research, an even better-performing network type or layout may exist for this task. It is also possible that similar methods could be adapted for effective use for $HH \to bb\tau\tau$ events with non-hadronically-decaying τ leptons, events in other HH decay channels, or even analyses that are not related to the Higgs boson self-coupling.

The neural network model predictions have already been shown to be capable of agreement between MC simulation and Run 2 ATLAS data in the analysis, suggesting that the neural network predicted m_{HH} could be used in full analyses in this channel. These model predictions have yet to be tested as input variables to the boosted decision trees used in full $HH \to bb\tau\tau$ analyses, so it is not yet known how significantly it may affect full analysis conclusions. Overall, the technique of using neural networks to regress the m_{HH} of $HH \to bb\tau\tau$ events has shown significantly improved performance compared to the MMC baseline and should be considered for use as an m_{HH} regression method in future ATLAS $HH \to bb\tau\tau$ analyses.

Bibliography

[1] Constraining the Higgs boson self-coupling from single- and double-Higgs production with the ATLAS detector using pp collisions at $\sqrt{s} = 13$ TeV. Technical report, CERN, Geneva, 2022. URL https://cds.cern.ch/record/2816332. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CONFNOTES/ATLAS-CONF-2022-050.

[2] LHC Machine. *JINST*, 3:S08001, 2008. doi: 10.1088/1748-0221/3/08/S08001.

[3] The ATLAS Collaboration et al. The ATLAS Experiment at the CERN Large Hadron Collider. *Journal of Instrumentation*, 3(08):S08003, aug 2008. doi: 10.1088/1748-0221/3/08/S08003. URL https://dx.doi.org/10.1088/1748-0221/3/08/S08003.

[4] ATLAS Collaboration. Search for resonant and non-resonant Higgs boson pair production in the $b\bar{b}\tau^+\tau^-$ decay channel using 13 TeV pp collision data from the ATLAS detector, 2022.

[5] ATLAS Collaboration. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys. Lett. B*, 716:1, 2012. doi: 10.1016/j.physletb.2012.08.020.

[6] CMS Collaboration. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Phys. Lett. B*, 716:30, 2012. doi: 10.1016/j.physletb.2012.08.021.

[7] M. Grazzini, G. Heinrich, S. Jones, S. Kallweit, M. Kerner, J. M. Lindert, and J. Mazzitelli. Higgs boson pair production at NNLO with top quark mass effects. *Journal of High Energy Physics*, 2018(5), may 2018. doi: 10.1007/jhep05(2018)059. URL https://doi.org/10.1007%2Fjhep05%282018%29059.

[8] R. L. Workman et al. Review of Particle Physics. *PTEP*, 2022:083C01, 2022. doi: 10.1093/ptep/ptac097.

[9] A. Elagin, P. Murat, A. Pranko, and A. Safonov. A new mass reconstruction technique for resonances decaying to $\tau\tau$. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 654(1):481–489, 2011. ISSN 0168-9002. doi: https://doi.org/10.1016/j.nima.2011.07.009. URL https://www.sciencedirect.com/science/article/pii/S0168900211014112.

[10] What are neural networks? URL https://www.ibm.com/topics/neural-networks.

[11] Keras. URL https://keras.io/.

[12] Jeff Backus. Investigating an RNN alternative to the MMC for m_{HH} estimation in $HH \to bb\tau_{\text{had}}\tau_{\text{had}}$ events with ATLAS. URL https://indico.cern.ch/event/1061300/contributions/4468779/attachments/2291462/3895992/rnn_mass_reco_slides.pdf.

[13] ATLAS open data 13 TeV documentation. URL http://opendata.atlas.cern/release/2020/documentation/index.html.

[14] Robin M. Schmidt. Recurrent neural networks (RNNs): A gentle introduction and overview, 2019.